Comparison of SRPT and PS Scheduling under ON/OFF load conditions

Nikhil Bansal Mor Harchol-Balter

Nov 2000

CMU-CS-00-179

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was supported by Cisco Systems via a grant from the Pittsburgh Digital Greenhouse 00-1 and by NSF ITR 99-167 ANI-0081396.

Keywords: overload, ON/OFF, scheduling, processor sharing, shortest processing remaining time, time-sharing, priority, unfairness, starvation, heavy-tailed behavior, high variance, single server, M/G/1.

Abstract

This paper analytically compares the performance of the SRPT (Shortest-Remaining-Processing-Time) and PS (Processor-Sharing) scheduling policies. SRPT scheduling has long been criticized for treating large jobs unfairly, whereas PS scheduling is by definition fair. We evaluate SRPT and PS under conditions where the unfairness under SRPT is believed to be most apparent: system overload. Specifically, we consider an single server with an alternating ON/OFF arrival process. During the ON periods the load at the server exceeds 1. During the OFF periods the load at the server is 0. We derive expressions for mean response time as a function of job size under PS and SRPT, for general job size distributions. In comparing these expressions, we find that for our ON/OFF model:

1. The mean response time under SRPT scheduling is far lower than under PS scheduling.
2. When the job size distribution is exponential, the biggest jobs may have higher mean response time under SRPT scheduling as compared with PS scheduling. However, when the job size distribution is heavy-tailed *all* jobs, including the very largest job, have lower (or only marginally higher) mean response times under SRPT scheduling as compared with PS scheduling. Heavy-tailed workloads are important because they arise naturally in many empirical computer workloads.

1 Introduction

In computer systems today, when multiple jobs contend for a single resource (e.g. CPU or bandwidth), the policy used for scheduling the jobs most closely resembles Processor-Sharing (PS). That is, the desired resource is time-shared among the contending jobs, with each job in turn receiving a small quantum of service.

It is well-known that the Shortest-Remaining-Processing-Time-First (SRPT) scheduling policy yields better mean response time than PS scheduling, however, applications have shied away from using this policy for fear that SRPT "starves" big jobs [2, 14, 15, 13]. The fear is that the response times of the big jobs will be much higher under SRPT scheduling than under PS scheduling.

In a recent paper [1], we proved that in the case of a single M/G/1 queue where the *job size* (processing requirement, or service requirement) distribution is heavy-tailed, this fear is unsubstantiated. We showed that, provided the system load ρ is such that $\rho < 1$ and ρ is not too close to 1, and provided the job size distribution is heavy-tailed, then *all* jobs, including the very largest job have lower mean response time under SRPT scheduling than under PS scheduling (and the overall mean response time is far lower under SRPT scheduling as compared with PS scheduling). The above analysis required that ρ not be too close to 1 (e.g. $\rho < .99$), so as to allow large jobs a turn to run. The above analysis was substantiated further by a second paper [6], where we implemented SRPT scheduling in a Web server, and compared performance of our SRPT-modified Web server with the original Web server in the case of $\rho < .95$ under trace-based Web workloads (which are heavy-tailed). (Web servers are a particularly good application for SRPT scheduling, since the service requirement of a Web request is proportional to the size of the file being requested, which is known) [6].

However, in real systems, such as Web servers, we can not be guaranteed that the load will always stay below 1. In fact it is much more common that the load bounces around, sometimes exceeding 1 and sometimes dropping to 0. Starvation under SRPT would appear to be more of a problem under overloaded conditions. The intuition commonly given is:

> *A big job arrives. Immediately thereafter a whole bunch of small jobs arrive, and the big job never gets to run under SRPT [2, 4, 14, 15, 13].*

The above statement would be obvious if the system remained in overload forever. However, it's not clear what exactly happens if the system alternates between overloaded and underloaded conditions (with mean system load under 1). Do big jobs necessarily do better under PS as compared with SRPT?

In this paper, we therefore consider an ON/OFF load model, where the arrival rate alternates between overload and zero load. The exact model is described in Section 2. For analytical tractability, we assume that the duration of the ON and OFF periods approach infinity. We assume a general job size distribution

Under the above model we derive the mean response time as a function of the job size under PS scheduling and under SRPT scheduling. We also derive several other interesting performance metrics, including the fraction of arriving jobs which are still in the system as a function of time, under SRPT and under PS scheduling. This latter metric has practical importance since it factors heavily into system overhead (e.g. context switching time and number of state buffers which must be maintained). We then evaluate these formulas in the case of a few specific job size distributions.

We find that when the job size distribution is *exponential*, SRPT improves over PS with respect to mean response time and number of jobs in the system by at least a factor of 4. However, this improvement comes at a price: the expected response time for large jobs under SRPT can be about 2-3 times higher than under PS. This "relative starvation" of the large jobs turns out to depend on the degree of overload during the ON period. The relative starvation of the big jobs under SRPT as compared with PS turns out to be worst when the load during the ON period is just slightly above 1.

However, when the job size distribution is *heavy-tailed*, which is characteristic of many computer workloads [11, 8, 5, 9, 12], the story changes. Under a Bounded Pareto job size distribution with α-parameter of 1.5, we find that SRPT improves over PS with respect to the mean response times of jobs and the number of jobs in the system by over an order of magnitude. These relative improvements of SRPT over PS are in fact greatest when the load during the ON period is not too high. Moreover, this improvement in mean performance *does not* come at the cost of hurting the performance of large jobs. In fact, even the very largest job has equal (or only marginally higher) response time under SRPT as compared with PS.

In Section 5 we discuss the above results in detail and provide intuition for why they hold. The above results are encouraging with respect to the potential real-world applicability of SRPT scheduling.

2 Problem Formulation and Notation

Throughout this paper we will be assuming an M/G/1 queue. Job sizes will be denoted by the random variable S. The job sizes will be assumed to be independent and identically distributed with c.d.f. $F(x)$ and p.d.f. $f(x)$.

The arrival process will consist of alternating ON/OFF periods. During the ON period (also called the "high load" period), jobs arrive with mean arrival rate λ_h and create a load of $\rho_h > 1$. The high load period has fixed duration t_h. During the OFF period (also called the "low load" period) jobs arrive with mean arrival rate λ_l and create a load of $\rho_l = 0$. The low load period has fixed duration t_l.

Let ρ denote the average system load. Thus

$$\rho = \frac{t_h}{t_h + t_l}\rho_h + \frac{t_l}{t_h + t_l}\rho_l$$

We will always assume that the average load $\rho < 1$.

Our model assumes alternating ON and OFF periods. During the ON period, jobs build up in the system as a function of time. We will derive an expression for this buildup. At the start of the OFF period, there is an accumulation of jobs which we refer to as "the bag of jobs." We will compute a distribution, F_r, on the remaining processing requirements of jobs in the bag at the start of the OFF period. The mean of this distribution will be $\mathbf{E}\{S_r\}$.

If we assume that t_h and t_l are finite, then we cannot guarantee that with probability 1 the number of jobs at the beginning of every ON period is zero. Furthermore, deriving the system state under finite t_h and t_l is extremely difficult. Thus we need to assume that $t_h \to \infty$ and $t_l \to \infty$. In this case it is easy to see by the Central Limit Theorem, that when the average load, $\rho < 1$, the number of jobs at the beginning of every ON period is zero with probability 1.

Some additional notation: The number of jobs in the system after t time into the high load period will be denoted by $N_h(t)$. Likewise the number of jobs in the system after t time into the low load period will be denoted by $N_l(t)$. Lastly, in analyzing SRPT scheduling, it will be useful to denote by $\rho(y)$ the load made up of jobs of size $< y$.

3 Previous Work

We have not found any literature discussing this problem prior to 1994. In 1994 Robert and Jean-Marie [10] analyzed the PS scheduling policy in the case of overload (not an ON/OFF process) and derived the number of jobs in the system as a function of t in the limit where $t \to \infty$. They also derived the limiting distribution of remaining times on jobs in the system in the case of $t \to \infty$. We use this result as a step within our proof of Theorem 1.

In 1997 Chen, Kella, and Weiss [3] analyzed the number of jobs in the system under the PS scheduling policy in the case of an OFF process which begins with a bag of jobs. Our goals differ from at of Chen et. al. in that we are concerned with response times rather than number of jobs in the system, and also, we are concerned with an ON/OFF workload. Our analytical methods differ from those used by Chen et. al..

We have not found any mention of SRPT analysis under overload or under our ON/OFF workload in the above papers or in any papers in the literature.

4 Analysis of response times under SRPT and PS

In this section we will derive formulas for the expected response time of a job of size x in the M/G/1/PS and M/G/1/SRPT systems under the ON/OFF load model.

We will start with a preliminary lemma, Lemma 1. Jean-Marie proved this lemma about

M/G/1/PS queues under overload. For completeness we give a quick overview of their proof.

Lemma 1 *Consider an M/G/1/PS queue with load $\rho > 1$ and average arrival rate λ. Let $N_h(t)$ denote the number of jobs in the system at time t. Then,*

$$\lim_{t\to\infty} \frac{N_h(t)}{t} = a \tag{1}$$

where a is the solution to the following equation:

$$1 - \int_0^\infty f(x)e^{-ax}dx = a/\lambda \tag{2}$$

Proof: Clearly $\lim_{t\to\infty} \frac{N_h(t)}{t} \leq \lambda$, so there exists b and c, such that for $t > c$, $N_h(t) \leq bt$.

Consider a job of size x, arriving at time $t_0 > c$, then the job departs at time t_d, such that

$$\begin{aligned} x &= \int_{t_0}^{t_d} \frac{dt}{N_h(t)} \\ &\geq \int_{t_0}^{t_d} \frac{dt}{bt} \\ &= \frac{1}{b} \log \frac{t_d}{t_0} \end{aligned}$$

So, $t_d \leq t_0 e^{bx}$. Thus, at time t, if a job of size x arriving at time $t_0 > c$ is present in the system, then certainly $t_0 \geq te^{-bx}$.

So, $N_h(t) \leq N_h(c) + \int_0^\infty \lambda(t - te^{-bx})f(x)dx$, provided $te^{-bx} > c$.

Choosing t large enough,

$$N_h(t) \leq N_h(c) + \lambda t(1 - L_f(b))$$

where $L_f(b)$ is the Laplace transform of f evaluated at b. Thus,

$$\frac{N_h(t)}{t} \leq \lambda(1 - L_f(b))$$

Let us choose $b = \lim\sup_{t\to\infty} \frac{N_h(t)}{t}$. Now observe that the function $h(x) = x - \lambda(1 - L_f(x))$ has a unique root a, in the range $x > 0$. Moreover, $h(x) \to \infty$ as $x \to \infty$. Thus, if $b > a$, then $(1 - L_f(b)) < b$, leading to a contradiction. Thus $b \leq a$.

Similarly we can show that $\lim\inf_{t\to\infty} \frac{N_h(t)}{t} \geq a$.

Thus the result follows. ■

Theorem 1 *In an M/G/1/PS system, under the ON/OFF load model, the expected response time for a job of size x, $\mathbf{E}\{T(x)_{PS}\}$ satisfies*

$$\lim_{t_h\to\infty} \frac{\mathbf{E}\{T(x)_{PS}\}}{t_h} = \frac{(1-e^{-ax})}{2} + (\rho_h - 1)F_{r_e}(x) - \frac{1}{2}(\overline{F}_r(x) - e^{-ax}) - \frac{\lambda}{2}\int_0^x \overline{F}(x-z)e^{-az}dz \tag{3}$$

where a is defined as in equation 2, and

$$\overline{F_r}(y) = \frac{\int_y^\infty \lambda f(z)(1-e^{-a(z-y)})dz}{a} \tag{4}$$

and

$$F_{r_e}(x) = \frac{a}{(\rho_h - 1)}\int_0^x \overline{F_r}(y)dy \tag{5}$$

Proof: Consider a job of size x which arrives into the system. A few obvious observations: First, the job must arrive somewhere during the ON period. Second, the job must complete before the next ON period begins. We therefore consider a single ON/OFF cycle beginning at time 0.

We can define some time, t_x during the ON period, such that the job of size x will complete before the ON period is over if and only if the job arrives before time t_x.

To compute t_x, observe that the service received by any job during the period $[t_x, t_h]$ equals x. Thus

$$\int_{t_x}^{t_h} \frac{dy}{N_h(y)} = x \tag{6}$$

Now by 1, $\lim_{t\to\infty} \frac{N_h(t)}{t} = a$. We will be loose and write this as $N_h(t) = at$, in which case 6 gives $t_x = t_h e^{-ax}$, or

$$\lim_{t_h\to\infty} \frac{t_x}{t_h} = e^{-ax} \tag{7}$$

A rigorous proof of 7 is given in the footnote below[1].

Henceforth, to keep the main idea clear, we will not distinguish between $\lim_{t_h\to\infty} \frac{t_x}{t_h} = e^{-ax}$ and $t_x = t_h e^{-ax}$. It can be seen that this will not matter, since we will eventually be concerned with the ratio of the response time and t_h. The arguments can be made rigorous as in the footnote 1.

[1] By Equation 1, $\lim_{t\to\infty} \frac{N_h(t)}{t} = a$. So, given an $\epsilon > 0$, $\exists C$, such that $|\frac{N_h(t)}{t} - a| < \epsilon$ for $t > C$. Assuming t_h is large enough that $t_h e^{-(a+\epsilon)x} > C$, we get that

$$t_h e^{-(a+\epsilon)x} \leq t_x \leq t_h e^{-(a-\epsilon)x}$$

Since $t_h \to \infty$, letting $\epsilon \to 0$, we get $\lim_{t_h\to\infty} \frac{t_x}{t_h} = e^{-ax}$.

We consider two cases: the case where the job of size x arrives prior to time t_x and the case where the job of size x arrives after time t_x. We will compute the mean response time in both cases.

First suppose that the job of size x arrives prior to time t_x. Let t_a denote the arrival time of the job, and let t_d denote its departure time (where $t_d < t_h$). Then by (1), $t_d = t_a e^{ax}$. So the response time of the job is

$$t_a \left(e^{ax} - 1\right) \tag{8}$$

Now since $t_h \to \infty$, we can assume that t_a is uniformly distributed in $[0, t_x]$, thus by (7) and (8) we have that:

$$\mathbf{E}\left\{T(x) | t_a \in [0, t_x]\right\} = \int_0^{t_x} \frac{y}{t_x} \left(e^{ax} - 1\right) dy = \frac{t_h}{2} \left(1 - e^{-ax}\right) \tag{9}$$

Now we approach the case where $t_a \in [t_x, t_h]$. In this case, our job of size x does not complete by time t_h. The remaining size on our job at time t_h will be denoted by x_r where

$$x_r = x - \int_{t_a}^{t_h} \frac{dy}{ay}$$

which evaluates to

$$x_r = x - \frac{1}{a} \log \frac{t_h}{t_a} \tag{10}$$

The remainder of this proof is devoted to computing the time from when the OFF period begins until our job completes.

We need some observations:

Consider an M/G/1/PS with load $\rho = 0$, starting at time 0 with an initial number of jobs $N_l(0)$, whose sizes are distributed with p.d.f. f_r.

Observation 1 *A job of size y can't depart until all jobs of size $< y$ have departed. So at the time when the job of size y departs, $N_l(0) \cdot F_r(y)$ jobs have departed.*

Observation 2 *When a job of size y departs, y units of work have been completed on all jobs of size $> y$.*

From the above two observations, it follows that our job of remaining size x_r will complete at time:

$$N_l(0) \int_0^{x_r} z f_r(z) dz + N_l(0) \overline{F_r}(z) x_r$$

This simplifies to

$$N_l(0)\mathbf{E}\{S_r\}F_{r_e}(x_r) \tag{11}$$

where $\mathbf{E}\{S_r\}$ represents the mean remaining size on jobs at time 0 and the $F_{r_e}(x_r)$ is the equilibrium distribution of F_r evaluated at x_r (i.e. $F_{r_e}(x_r) = \frac{1}{E[S_r]}\int_0^{x_r}\overline{F}_r(y)dy$).

Now observe that $N_l(0)\mathbf{E}\{S_r\}$ is just the work in the system at time 0, which is equal to $(\rho_h - 1)t_h$. Thus, the time from the start of the OFF period until our job of remaining size x_r completes is:

$$(\rho_h - 1)t_hF_{r_e}(x_r)$$

So the response time of our job is:

$$t_h - t_a + (\rho_h - 1)t_hF_{r_e}(x_r) \tag{12}$$

Now, given that t_a is uniformly distributed in $[t_x, t_h]$, by (10) and (12) we have that,

$$\mathbf{E}\{T(x)|t_a \in [t_x, t_h]\} = \int_{t_x}^{t_h} \frac{1}{(t_h - t_x)}\left((t_h - y) + (\rho_h - 1)t_hF_{r_e}\left(x - \frac{1}{a}\log\frac{t_h}{y}\right)\right)dy \tag{13}$$

Observe however that we have not yet derived F_{r_e}. That is, we still need to compute F_r, the distribution on the remaining sizes of jobs at the beginning of the OFF period. We do this now:

We will compute the number of jobs with remaining size $> y$ at the moment that the OFF period begins.

Consider a job of (original) size z. This job will have remaining size $> y$ at time t_h iff it's arrival time, t_a, is such that $t_a > t_he^{-a(z-y)}$ (by the above type of arguments).

Thus the total number of jobs of size $(z, z + dz)$ which have remaining size $> y$ at time t_h is the number of jobs which arrive during $t_he^{-a(z-y)}$ and t_h, which is

$$\lambda t_hf(z)(1 - e^{-a(z-y)})dz \tag{14}$$

Therefore the total number of jobs which have remaining size $> y$ at time t_h is obtained by integrating 14 over all possible job sizes (greater than y), thus,

$$\int_y^{\infty} \lambda t_hf(z)(1 - e^{-a(z-y)})dz \tag{15}$$

To obtain $\overline{F}_r(y)$ observe that the total number of jobs at time t_h is at_h. Thus, using 15 the fraction of jobs with remaining size $> y$ at the start of the OFF period will be given by

$$\overline{F_r}(y) = \frac{\int_y^{\infty}\lambda f(z)(1 - e^{-a(z-y)})dz}{a} \tag{16}$$

Moreover,

$$F_{r_e}(x) = \frac{1}{E[S_r]} \int_0^x \overline{F_r}(y)dy \tag{17}$$

To calculate $E[S_r]$, observe that $E[S_r]$ is equal to the ratio of the total remaining work and the total number of remaining jobs at the beginning of the OFF period. Since the total remaining work at the end of the OFF period is $(\rho_h - 1)t_h$, and the number in system is by definition at_h,

$$E[S_r] = \frac{(\rho_h - 1)}{a} \tag{18}$$

Thus, we get

$$F_{r_\epsilon}(x) = \frac{a}{(\rho_h - 1)} \int_0^x \overline{F_r}(y)dy \tag{19}$$

Finally, since

$$\mathbf{E}\left\{T(x)\right\} = \frac{t_x}{t_h}\mathbf{E}\left\{T(x)|t_a \in [0, t_x]\right\} + \frac{t_h - t_x}{t_h}\mathbf{E}\left\{T(x)|t_a \in [t_x, t_h]\right\}$$

Substituting 9 and 13 we get,

$$\mathbf{E}\left\{T(x)\right\} = \frac{t_x}{t_h}\cdot\frac{t_h}{2}(1-\epsilon^{-ax})+\frac{t_h - t_x}{t_h}\cdot\int_{t_x}^{t_h} \frac{1}{(t_h - t_x)}\left((t_h - y) + (\rho_h - 1)t_h F_{r_e}\left(x - \frac{1}{a}\log\frac{t_h}{y}\right)\right) dy \tag{20}$$

$$\mathbf{E}\left\{T(x)\right\} = \frac{t_x}{2}(1 - \epsilon^{-ax}) + \frac{1}{t_h}\left(\frac{(t_h - t_x)^2}{2} + \int_{t_x}^{t_h} (\rho - 1)t_h F_{r_e}\left(x - \frac{1}{a}\log\frac{t_h}{y}\right) dy\right)$$

$$\mathbf{E}\left\{T(x)\right\} = \frac{(t_h - t_x)}{2} + \int_{t_x}^{t_h} (\rho - 1)F_{r_e}\left(x - \frac{1}{a}\log\frac{t_h}{y}\right) dy \tag{21}$$

where

$$F_{r_e}(y) = \frac{1}{(\rho - 1)} \int_0^y \int_w^\infty \lambda f(z)(1 - \epsilon^{-a(z-w)})dz dw$$

Observe that the expression for $E\{T(x)\}$ in (21) involves three nested integrals, which makes it difficult to evaluate using a symbolic math package like $Mathematica^{TM}$. After some algebraic manipulation the number of nested integrals can be reduced to two. The proof is given in the Appendix. Finally we get,

$$\mathbf{E}\left\{T(x)\right\} = \frac{(t_h - t_x)}{2} + (\rho - 1)F_{r_e}(x)t_h - \frac{t_h}{2}(\overline{F}_r(x) - \epsilon^{-ax}) - \frac{\lambda t_h}{2}\int_0^x \overline{F}(x - z)\epsilon^{-az}dz$$

■

Lemma 2 *Let x_o be defined such that $\rho_h(x_o) = 1$. Under SRPT, as $t_h \to \infty$, the rate at which jobs complete during the high load period is $\lambda F(x_o)$ (and the rate of growth of jobs in the system is thus $\lambda \overline{F}(x_o)$). The p.d.f. of job sizes remaining in the system is:*

$$f_r(x) = \frac{f(x)}{\overline{F}(x_o)}, \quad \textit{where } x > x_o$$

Proof: Consider the high load period. Let $W(x,t)$ denote the work made up by jobs of of size less than x, arriving by time t. A necessary condition for a job of size x to receive service at time t is that $W(x,t) - t < 0$.

For a job of size $x > x_o$, the expected value of $W(x,t)$ is $\rho(x)t$. Thus, by the Central Limit Theorem, $W(x,t)$ is distributed like $\rho(x)t + N\sqrt{t}$, where N is some normally distributed random variable. As $t_h \to \infty$ we see that $W(x,t) > t$ with probability 1. Thus the fraction of jobs with size x or greater which receive service during the high load period goes to 0. Thus the result follows.

■

Theorem 2 *In an M/G/1/SRPT system, under the ON/OFF load model, the expected response time for a job of size x is*

$$\lim_{t_h \to \infty} \frac{\mathbf{E}\{T(x)\}_{SRPT}}{t_h} = 0, \quad if \quad x < x_o$$

$$\lim_{t_h \to \infty} \frac{\mathbf{E}\{T(x)\}_{SRPT}}{t_h} = \rho_h(x) - \frac{1}{2}, \quad if \quad x > x_o$$

where x_o is such that $\rho_h(x_o) = 1$.

Proof: For jobs of size less than x_o, clearly the response time is independent of t_h. So

$$\lim_{t_h \to \infty} \frac{E[T(x)]_{SRPT}}{t_h} = 0$$

for $x < x_o$.

By Lemma 2, with probability 1, a job of size greater than x_o receives service only during the OFF period. So, the expected waiting time of the job during the ON period will be $\frac{t_h}{2}$. Moreover, the time since the beginning of the OFF period until this job is serviced is equal to the work made up by jobs of size between x_o and x. This amount of work is equal to $\int_{x_o}^{x} \lambda t_h y f(y) dy$, which is equal to $(\rho_h(x) - \rho_h(x_o))t_h$, or just $(\rho_h(x) - 1)t_h$, since $\rho_h(x_o) = 1$.

Thus it follows that the mean response time for a job of size x, such that $x > x_o$ will be $\frac{t_h}{2} + (\rho_h(x) - 1)t_h$. ■

5 Direct analytic comparison of SRPT and PS under various workloads

In Section 4 we derived the expression for the expected response time of a job of size x under SRPT and PS for the ON/OFF model. In this section we investigate these results for specific job size distributions. We investigate 2 different types of distributions:

1. Exponential.
2. Bounded Pareto, $B(\alpha = 1.5)$. This is a heavy-tailed distribution.
   Recall a *Pareto* distribution with parameter α, is defined such that
   $$Pr[X > x] \sim x^{-\alpha}.$$
   The *Bounded-Pareto* distribution [7] is characterized by three parameters: α, the exponent of the power law; k, the smallest possible job; and p, the largest possible job, The probability density function for the Bounded Pareto $B(k, p, \alpha)$ is defined as:
   $$f(x) = \frac{\alpha k^\alpha}{1 - (k/p)^\alpha} x^{-\alpha-1} \quad k \leq x \leq p.$$
   In this paper, $B(\alpha)$ will denote the distribution $B(k, p, \alpha)$ obtained by keeping the mean fixed (at 3000) and the maximum value fixed (at $p = 10^{10}$), which correspond to typical values for Web workloads taken from [5].

5.1 Normalized Mean Response Times as a function of job size

We begin by focusing our discussion on two performance metrics:

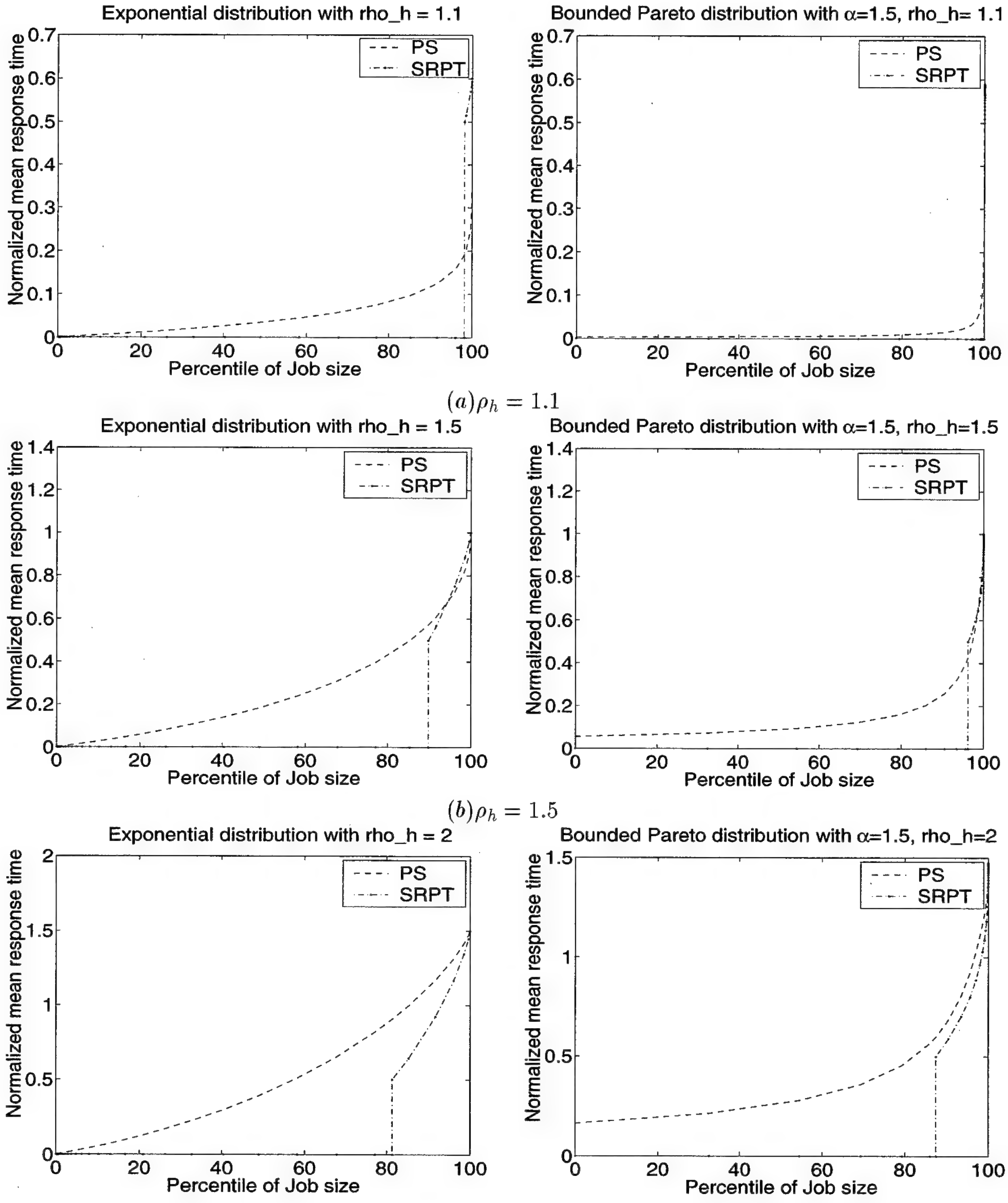
1. Mean response time (under SRPT versus under PS).
2. Expected response time for large jobs. Specifically we will be interested in whether large jobs "starve" under SRPT scheduling as compared with PS scheduling.

Consider the mean response time for a job of size x, $\mathbf{E}\{T(x)\}$. Under our ON/OFF load model, $\mathbf{E}\{T(x)\}$ is proportional to t_h. Thus rather than discuss mean response time, we instead show *normalized mean response time*, defined as follows:[2]

$$\text{Normalized Mean Response Time for job of size } x = lim_{t_h \to \infty} \frac{\mathbf{E}\{\text{T(x) | ON period of length } t_h\}}{t_h}$$

Figure 1 shows the normalized mean response time as a function of job size under SRPT scheduling versus PS scheduling. The job size is expressed as a percentile of the job size

[2] In the definition for normalized mean response time, we assume that the length of the OFF period is ∞.



$(a)\rho_h = 1.1$

$(b)\rho_h = 1.5$

$(c)\rho_h = 2$

Figure 1: *This figure shows the normalized mean response times for various values of ρ_h ($\rho_h = 1.1, \rho_h = 1.5$ and $\rho_h = 2$). The first column assumes an exponential job size distribution s and the second column assumes the $B(1.5)$ distribution.*

distribution (where 100 percentile indicates the very largest job). Observe furthermore that due to the choice of the x-axis, the area under the PS (respectively, SRPT) curve corresponds to the normalized mean response time under PS (respectively, SRPT). The plots on the left column of Figure 1 considers the exponential job size distribution at various values of ρ_h, and the plots on the right consider the distribution $B(\alpha = 1.5)$.

We begin with a few observations which hold for all plots and are easily explained:

Observation 3 *Under SRPT most of the jobs have a normalized response time of 0.*

To see why, observe that under SRPT a job of size x will have a normalized response time of 0, if $\rho_h(x) < 1$, that is if the load made up by jobs of size $< x$ during the ON period is less than 1. However under PS, every job has a non-zero normalized response time.

Observation 4 *Under SRPT, let x be the size of the smallest job which has a non-zero normalized response time. Observe that job x always seems to have a normalized response time of exactly $\frac{1}{2}$.*

To see why, observe at job x never gets to run during the ON period and thus has an average waiting time of $t_h/2$ during the ON period. Moreover, job x gets to run immediately when the OFF period begins. Thus its normalized response time (as $t_h \to \infty$) is 1/2.

Observation 5 *Under SRPT, the normalized response time approaches $\rho_h - \frac{1}{2}$ for large jobs.*

To see why, observe that a large job has a waiting time of $\frac{t_h}{2}$ on the average during the ON period, and it receives service only towards the end during the OFF period. Since it takes $(\rho_h - 1)t_h$ time to finish off accumulated work, the result follows.

We now walk through Figure 1 and use the above observations to explain the plots. First compare Figure 1a (left) with Figure 1a (right). Both assume $\rho_h = 1.1$, but Figure 1a(right) assumes an exponential job size distribution, while Figure 1a(left) assumes a $B(\alpha = 1.5)$ distribution. Let us contrast the curves with respect to the performance of the big jobs. Under the exponential job size distribution, jobs in the 0–98.5 percentile have a normalized response time of 0 under SRPT. The job in the 98.5^{th} percentile has a normalized response time of 0.5 under SRPT, but only 0.18 under PS. In fact, almost all the jobs in the top 1.5 percentile do about 2-3 times worse under SRPT as compared with PS under the exponential job size distribution. By contrast under the $B(1.5)$ distribution (though not very clear from the plot) the normalized response under SRPT becomes non-zero only for jobs in the 99.94 or higher percentile. This is true because $B(1.5)$ is more heavy tailed than the exponential distribution, and thus only .06% of the large jobs account for the load between 1 and 1.1. Furthermore, while a job in the 99.94 percentile has a normalized response time of 0.5 under

SRPT, it has a normalized response time of almost 0.4 under PS. Thus the performance for this job under PS and SRPT is comparable. In fact, for slightly bigger jobs than this, the response times under PS become even worse than that under SRPT.

Now contrast Figure 1a(right) and Figure 1a(left) with respect to mean normalized response time. For both figures the area under the SRPT curve is smaller than the area under the PS curve, — thus the mean response time under SRPT is much lower than under PS. The difference in mean response time is magnified in the case of the $B(1.5)$ workload, because almost all jobs have zero normalized time under SRPT (and all remaining jobs are comparable under SRPT and PS).

Figure 1b shows the normalized mean response time under $\rho_h = 1.5$. Observe that for both the job size distributions there is negligible starvation for large jobs under SRPT as compared with PS. Arguing as in the above paragraph, for both job size distributions, the mean normalized response time is far lower under SRPT than under PS. Furthermore, this difference is exaggerated in the $B(1.5)$ distribution where under SRPT only 2.8% jobs have a non-zero normalized response time, as opposed to about 10% for the exponential distribution.

Finally when $\rho_h = 2$ (Figure 1c), we observe that there is no starvation at all under SRPT as compared with PS under either job size distribution. In fact every job seems to perform worse (or similar) under PS, as compared to SRPT. This may seem contradictory since one could argue that at least the very largest jobs should have a larger response time under SRPT than under PS. We will explain this below.

We summarize the trends we have seen in three observations. We offer intuition explaining each observation.

Observation 6 *Large jobs do not necessarily suffer under SRPT as compared with PS (as is commonly believed). (See Figure 1c.) Particularly for heavy-tailed distributions, large jobs often do at least as well under SRPT as compared with PS, with respect to their expected response time. (See Figure 1 whole right column).*

To see why this is the case, observe that although large jobs do badly under SRPT, they do *almost equally badly* under PS. The point is that the average amount of service received by a large job during an ON period is negligible compared to its size. Thus this job stays in the system throughout the ON period (since its arrival). Moreover it is among the last of the jobs to complete during the OFF period, since its remaining size at the beginning of the OFF period is large compared to other remaining jobs.

Observation 7 *For a fixed ρ_h, the more heavy-tailed the job size distribution the better the performance of big jobs under SRPT as compared with PS. Also, the more heavy-tailed the job size distribution the greater the improvement in the normalized mean response time under SRPT as compared with PS. (See Figure 1 right column as compared with left column).*

To see why this is the case, recall that in a more heavy-tailed distribution, there are fewer jobs comprising the excess load. These few jobs are very big (think of them as elephants). Now, by the same argument as above, these elephants don't make much progress under PS during the ON period, thus their response times are comparable under PS and under SRPT. Furthermore, all jobs other than the elephants experience zero normalized mean response time under SRPT. Thus the overall normalized mean response time under SRPT is much lower than under PS.

Observation 8 *The higher the value of ρ_h, the lower the starvation of big jobs under SRPT as compared with PS. (See Figure 1)*

This is a surprising observation, in that one might conjecture that a *low* value of ρ_h (barely overloaded ON times) would yield less starvation of big jobs. However it turns out that the reverse is true. Here's some intuition: When ρ_h is closer to 1, PS gets more work done on the big jobs during the ON period. Thus at the start of the OFF period, the response times of the big jobs under PS is *much* smaller. Thus the big jobs in SRPT *appear* to be "starved" by comparison, although in fact their response times have improved as well — but not by as much. This intuition is in complete agreement with Figure 1.

5.2 Growth in number of jobs in the system over time

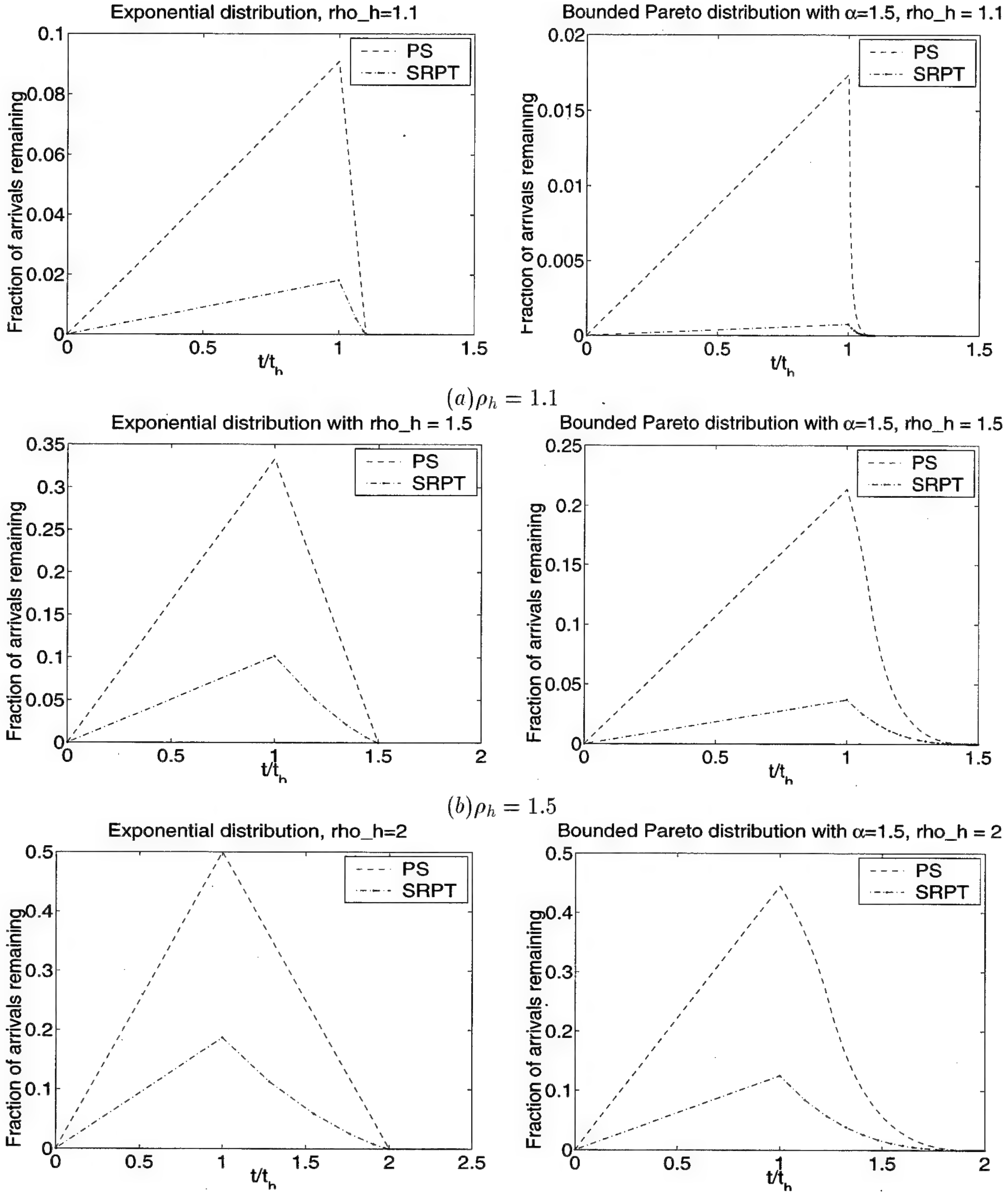
We now consider the number of jobs in the system as a function of time under PS and SRPT. The number in system is an interesting practical metric. Consider as an example a Web server which services its requests in SRPT order, as opposed to the traditional PS service order. The number of requests in the system corresponds to the number of simultaneously open connections in the Web server. The greater this number the more overhead is required by the Web server. Furthermore, if this number gets too high, the Web server simply crashes.

The mean number of jobs in the system is obviously an increasing function of t_h. Thus we instead look at the *fraction of arrivals remaining in the system*, which is defined as:

$$\text{Fraction of arrivals remaining} = lim_{t_h \to \infty} \frac{\mathbf{E}\{\text{N(t)} \mid \text{ON period of length } t_h\}}{\lambda t_h}$$

where $N(t)$ denotes the number of jobs in the system at time t. We consider a normalized time axis, showing t/t_h, rather than t.

Figure 2 shows the fraction of arrivals remaining as a function of normalized time, for various values of ρ_h, under SRPT and PS scheduling. Observe, a value of 1 on the x-axis indicates the end of an ON period. The plots in the left column assume an exponential job size distribution, whereas the plots in the right column assume the distribution $B(1.5)$. *Again, the area under the curves is proportional to the mean number of jobs in the system (hence to the mean response time).*



(a) $\rho_h = 1.1$

(b) $\rho_h = 1.5$

(c) $\rho_h = 2$

Figure 2: *This figure shows the fraction of arrivals remaining as a function of normalized time under (a) $\rho_h = 1.1$, (b) $\rho_h = 1.5$ and (c) $\rho_h = 2$. The left column corresponds assumes an exponential job size distribution, and the right column a heavy-tailed $B(1.5)$ distribution.*

Below we state a few observations about the graphs and provide intuition explaining these observations.

Observation 9 *For all plots in Figure 2, the number of jobs in the system under SRPT is always significantly less than that under PS.*

Observation 10 *For a fixed ρ_h, the more heavy-tailed the job size distribution the greater the improvement in number of jobs in the system under SRPT as compared with PS. (See Figure 2 right column as compared with left column).*

For example, Figure 2a (left) shows the number in system for the exponential distribution when $\rho_h = 1.1$. Observe that the number in system under SRPT is always 4-5 times lesser than that under PS. Figure 2a (right) shows the number is system for $B(1.5)$ distribution when $\rho_h = 1.1$. The improvements in the mean number in system under SRPT are much more significant, about 15-20 times better than that under PS. This observation coupled with Observation 6 about starvation in Figure 1a (right) makes a very strong case for SRPT. *Not only is there a 15-20 times improvement in the mean response time under SRPT, but this improvement does not come at the cost of starving large jobs.*

Observation 9 makes sense since SRPT obviously minimizes the number of jobs in the system at any time. The intuition behind Observation 10 is the same as that behind Observation 7.

Observation 11 *Though still significant, the relative advantage of SRPT over PS (with respect to number of jobs in the system) decreases at higher values of ρ_h. (See Figures 2b and 2c).*

Observe that this does *not* contradict Observation 8, since that observation is concerned only with the large jobs.

Observation 12 *The number in system increases linearly during the ON period for both SRPT and PS.*

This follows from Lemma 1 and Lemma 2.

Observation 13 *The curve for the number of jobs under SRPT during the OFF period is always convex.*

This follows since SRPT works on jobs with the smallest remaining size first, thus the rate of clearance of jobs is maximum in the beginning of the OFF period and then decreases.

Observation 14 *Observation 13 above is not true for PS. In fact, since PS timeshares among all the jobs, it somewhat delays getting jobs out at the beginning of the OFF period. This can be observed significantly in Figure 2c (right). Thus SRPT not only accumulates fewer jobs, but it also gets them out as quickly as possible.*

We end with an interesting side note:

Observation 15 *The number in system under PS decreases linearly under PS for the exponential distribution.*

To see this, observe that at the end of an ON period, the remaining sizes of the jobs are also exponentially distributed with some rate, say μ,(since the distribution of the remaining size of a job conditional on the fact that it has received x amount of service is still exponential). Now observe that regardless of how many jobs there are remaining at the end of the ON period, the rate at which jobs will compete under PS is a constant: μ. (This follows, since if there are n jobs in the system, then since each job receives $\frac{1}{n}^{th}$ of the service, it is likely to finish with rate $\frac{\mu}{n}$. Since there are n jobs, the total rate at which jobs leave the system will be $n \cdot \frac{\mu}{n} = \mu$.)

6 Conclusion

This paper examines a load model consisting of alternating ON/OFF periods where the system load exceeds 1 during the ON period and is 0 during the OFF period. Under this load model, we compare SRPT and PS scheduling. We find that SRPT scheduling is a big win with respect to mean performance metrics like mean response time and mean number of jobs in the system. More surprisingly, we find that, in the case where the job size distribution is *heavy-tailed*, this win does not come at the cost of starving the large jobs. In fact *all jobs* including the very largest perform better or only marginally worse under SRPT scheduling, as compared with PS scheduling.

The above results seem counterintuitive. Shouldn't large jobs starve more under SRPT given temporary overload? We've found that the answer is no, but the reason can only partially be attributed to the superior properties of SRPT. At least equally relevant is the poor performance of PS. Our analysis shows that PS is particularly ineffective in dealing with periods of temporary overload. Due to its time-sharing nature, it deteriorates the performance of all the jobs. Moreover, PS is particularly slow at getting the system "back to normal" once the overload has disappeared. By contrast, SRPT accumulates far fewer jobs during the overload period, and is also much more efficient at getting them out once the overload period is over. A heavy-tailed job size distribution works strongly in SRPT's favor because it allows SRPT to complete all but a small fraction of the jobs during the overload period.

The superior performance of SRPT under fluctuating load conditions and heavy-tailed job sizes make SRPT an attractive alternative to the more traditional PS scheduling in applications where job sizes are known, or can be estimated.

References

[1] Nikhil Bansal and Mor Harchol-Balter. Analysis of SRPT scheduling: Investigating unfairness. Technical Report CMU-CS-00-149, CMU Computer Science Department, July 2000.

[2] M. Bender, S. Chakrabarti, and S. Muthukrishnan. Flow and stretch metrics for scheduling continous job streams. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[3] H. Chen, O. Kella, and G. Weiss. Fluid approximations for a processor sharing queue. *Queueing Systems: Theory and Applications*, 27:99–125, 1997.

[4] E.G. Coffman and L. kleinrock. Computer scheduling methods and their countermeasures. In *AFIPS conference proceedings*, volume 32, pages 11–21, 1968.

[5] M. Crovella and A. Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. In *Proceedings of the 1996 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 160–169, May 1996.

[6] Mor Harchol-Balter, Nikhil Bansal, and Bianca Schroeder. Implementation of SRPT scheduling in web servers. Technical Report CMU-CS-00-170, CMU Computer Science Department, Nov 2000.

[7] Mor Harchol-Balter, Mark Crovella, and Cristina Murta. On choosing a task assignment policy for a distributed server system. *IEEE Journal of Parallel and Distributed Computing*, 59:204 – 228, 1999.

[8] Mor Harchol-Balter and Allen Downey. Exploiting process lifetime distributions for dynamic load balancing. In *Proceedings of SIGMETRICS '96*, pages 13–24, 1996.

[9] G. Irlam. Unix file size survey - 1993. Available at `http://www.base.com/gordoni/ufs93.html`, September 1994.

[10] A. Jean-Marie and P. Robert. On the transient behavior of the processor-sharing queue. *Queueing Systems: Theory and Applications*, 17:129–136, 1994.

[11] W. E. Leland and T. J. Ott. Load-balancing heuristics and process behavior. In *Proceedings of Performance and ACM Sigmetrics*, pages 54–69, 1986.

[12] David L. Peterson and David B. Adams. Fractal patterns in DASD I/O traffic. In *CMG Proceedings*, December 1996.

[13] A. Silberschatz and P. Galvin. *Operating System Concepts, 5th Edition.* John Wiley & Sons, 1998.

[14] W. Stallings. *Operating Systems, 2nd Edition.* Prentice Hall, 1995.

[15] A.S. Tanenbaum. *Modern Operating Systems.* Prentice Hall, 1992.

7 Appendix

We will simplify the integral $I = \int_{t_x}^{t_h} F_{r_e}\left(x - \frac{1}{a}\log\frac{t_h}{y}\right) dy$

Set $y = t_h e^{-az}$, then $dy = -a t_h e^{-az} dz$ and

$$I = \int_0^x F_{r_e}(x - z) a t_h e^{-az} dz$$

Applying integration by parts, we get

$$I = [-F_{r_e}(x-z)t_h e^{-az}]_0^x - \int_0^x t_h e^{-az}\overline{F_r}(x-z)\frac{1}{E[S_r]}dz$$

$$I = F_{r_e}(x)t_h - I_1$$

where

$$I_1 = \int_0^x t_h e^{-az}\overline{F_r}(x-z)\frac{1}{E[S_r]}dz \tag{22}$$

We will show in Lemma 4 that

$$I_1 = \frac{t_h}{(2a)E[S_r]}(\overline{F}_r(x) - e^{-ax}) + \frac{\lambda t_h}{2aE[S_r]}\int_0^x \overline{F}(x-z)e^{-az}dz$$

Thus giving us

$$I = F_{r_e}(x)t_h - \frac{t_h}{(2a)E[S_r]}(\overline{F}_r(x) - e^{-ax}) + \frac{\lambda t_h}{2aE[S_r]}\int_0^x \overline{F}(x-z)e^{-az}dz$$

We first begin with an identity which relates the pdf and the cdf of the remaining service times of jobs under PS.

Lemma 3 *Let $f_r(y)$ and $F_r(y)$ denote respectively the pdf and cdf of the remaining service of the jobs at the start of the OFF period under PS. Then*

$$\overline{F}_r(y) = \frac{\lambda\overline{F}(y)}{a} - \frac{f_r(y)}{a}$$

Proof: By (16) we know that

$$\begin{aligned}\overline{F_r}(y) &= \frac{\int_y^\infty \lambda f(z)(1-e^{-a(z-y)})dz}{a} \\ &= \frac{\lambda\overline{F}(y)}{a} - e^{ay}\int_y^\infty \frac{\lambda}{a}f(z)e^{-az}dz \end{aligned}\tag{23}$$

Since $f_r(y) = -\frac{d}{dy}\overline{F}_r(y)$

$$-f_r(y) = \frac{-\lambda}{a}f(y) + e^{ay}\frac{\lambda}{a}f(y)e^{-ay} - ae^{ay}\int_y^\infty \frac{\lambda}{a}f(z)e^{-az}dz$$

Observing that the first and second terms cancel out, we get

$$f_r(y) = \lambda e^{ay}\int_y^\infty f(z)e^{-az}dz \tag{24}$$

Comparing the expressions for $\overline{F}_r(y)$ and $f_r(y)$ given by (23) and (24), the result follows. ■

Lemma 4 *Let I_1 be defined as in equation 22, then*

$$I_1 = \frac{t_h}{(2a)E[S_r]}(\overline{F}_r(x) - e^{-ax}) + \frac{\lambda t_h}{2aE[S_r]}\int_0^x \overline{F}(y)e^{-az}dz$$

Proof:

$$\begin{aligned}
I_1 &= \int_0^x \frac{t_h e^{-az}}{E[S_r]}\overline{F}_r(x-z)dz \\
&= \left[-\overline{F}_r(x-z)\frac{t_h}{aE[S_r]}e^{-az}\right]_0^x - \int_0^x -e^{-az}\frac{t_h}{aE[S_r]}\cdot(-f_r(x-z))\cdot -dz \\
&= \frac{t_h}{aE[S_r]}(\overline{F}_r(x) - e^{-ax}) + \int_0^x e^{-az}\frac{t_h}{aE[S_r]}\left(\lambda\overline{F}(x-z) - a\overline{F}_r(x-z)\right)dz \text{ (By Lemma 3)} \\
&= \frac{t_h}{aE[S_r]}(\overline{F}_r(x) - e^{-ax}) + \frac{\lambda t_h}{2aE[S_r]}\int_0^x \overline{F}(x-z)e^{-az}dz - I_1
\end{aligned}$$

The last equality follows, since the final term in this expression is just I_1. Thus

$$I_1 = \frac{t_h}{(2a)E[S_r]}(\overline{F}_r(x) - e^{-ax}) + \frac{\lambda t_h}{2aE[S_r]}\int_0^x \overline{F}(x-z)e^{-az}dz$$

■